ElectioNet™

State of New Jersey Office of the Attorney General Division of Elections

Development Milestones/Plan for the Statewide Voter Registration System (SVRS)

NEW JERSEY

Deliverable SVRS 058

Presented to:
Michael Gallagher
SVRS Project Manager
Department of Law and Public Safety
Trenton, New Jersey

Presented by:
Covansys Corporation
32605 West 12 Mile Road
Farmington Hills, MI 48334

June 2005

Revision History

Date	Brief Description	Changed By:
06/30/2005	Version 1	Bob Brandner
06/30/2005	Final Formatting	Wm. Gary Bush

Table of Contents

INTRODUCTION

Methodology

The Electio*Net* product was developed using the Rational Unified Process (RUP) methodology. This methodology establishes and enforces formal processes and management disciplines throughout the development cycle from the initial requirements gathering through the final phase of User Acceptance Testing. This methodology is particularly well suited to guiding and controlling the development of applications like ElectioNet that utilize component-based modules

This document provides a detailed description of the Development Plan, Methodology, and associated Milestones for modifying the COTS-version of Electio*Net* to meet the state-specific requirements for the NJ SVRS.

Consistent with the diagram that follows, the ElectioNet software modification process is highlighted by the development milestones listed in sequence below:

1. Collection of Requirements

- Joint Application Development (JAD) sessions with user representatives to detail needs.
- Preparation of Gap documents to determine alterations/additions to the COTS product.

2. Analysis and Design

- Summary of all individual requirements with tracking to their source (Requirements Traceability Matrix).
- Summary of the functional requirements (FRD's) … i.e., how things work.
- Review and Approval of the requirements set by the State's core review team.

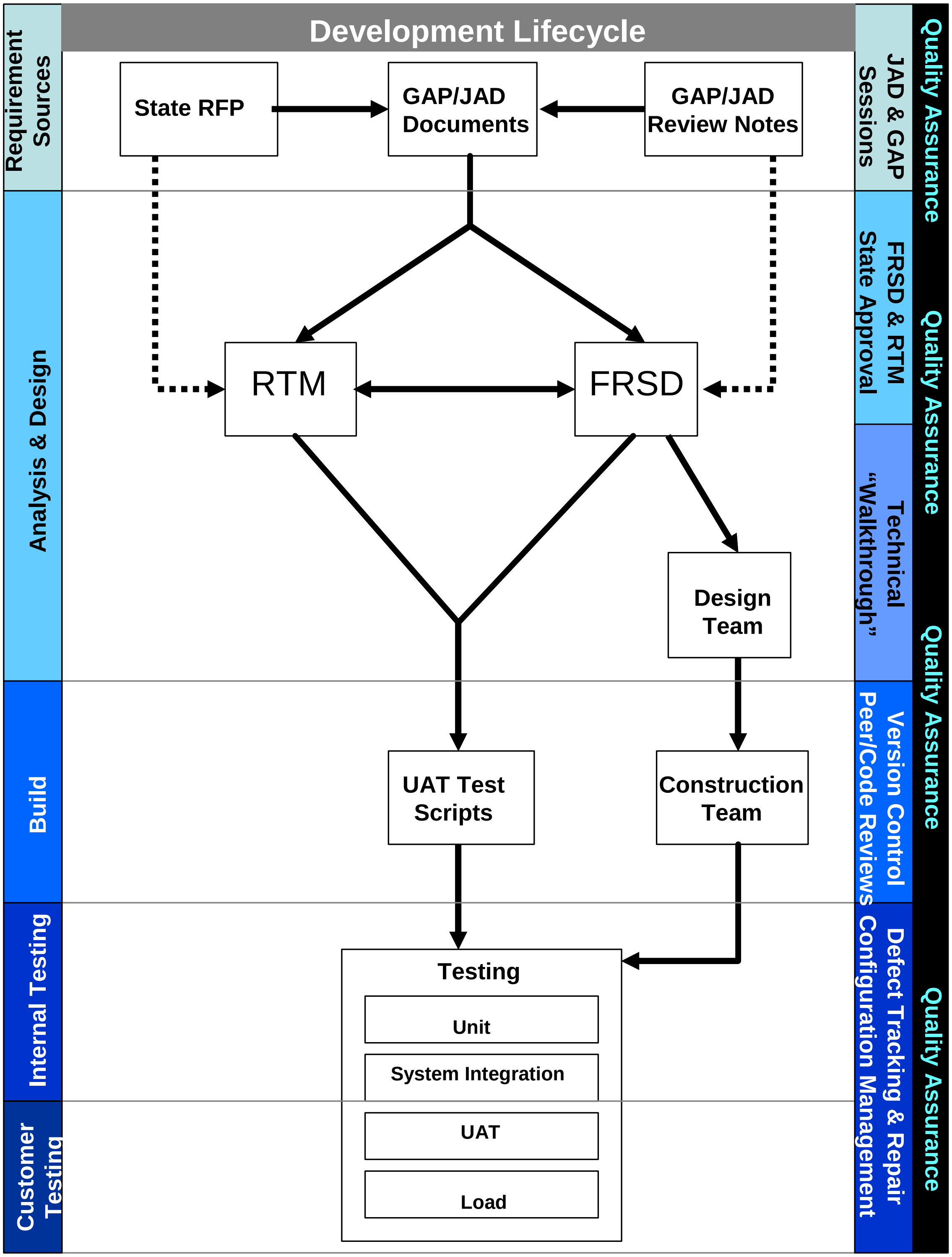
3. Application Build/Construction

- Modification of the COTS application to include all New Jersey requirements.
- Preparation of test scripts to ensure all functionality and requirements can be exercised in the modified application.

4. Testing – Internal & External

- Internal testing to verify completeness of the modification and elimination of errors.
- User Acceptance and Performance Testing by the State's testers to verify conformance to the specifications.

It would be remiss not to mention that the integrity of the entire development process is achieved and maintained by interweaving the disciplines of Quality Assurance and Configuration Management into every task. This document starts with an explanation of how these disciplines will be applied during this project.



QUALITY ASSURANCE

Overview

Covansys/Covansys Quality Assurance (QA) is an integral component of the delivery approach. Its primary purpose is to ensure that each aspect of the engagement meets the client's objectives. QA has found that without a specific focus on measurable quality and productivity, results will vary across project areas and project activities. Therefore, QA activities focus on improving project productivity, reducing costs, improving project quality, and measuring compliance with engagement objectives and progress.

The intent of the QA process is to provide project teams with procedures and standards that maximize quality, and with direct performance feedback, allowing them to identify and resolve problems. In QA, everyone working on a project is responsible for quality assurance.

At the project's inception, the QA Team works with the New Jersey SOS to ensure implementation of the Quality Assurance processes. Our approach to QA includes the following techniques:

- Standards definition – document and mutually agree upon standards, such as formats and content of project documents.
- Planning reviews – review associated project objectives and constraints for each task area and recommend tools and techniques, methodologies, project infrastructure, and performance measures and targets.
- Deliverable reviews – each deliverable will undergo a review cycle to ensure that the appropriate audience has had the opportunity to review and comment on the document.

The institution of these techniques, in conjunction with our project management approach, ensures that the New Jersey SVRS project completely addresses the requirements, correctly solves the business and technical challenges presented, applies standards and reusable components consistently, and provides a clear communication with end users and technical staff for evaluation of work products.

As consistent high quality in our products and services is critical to helping our clients achieve their business objectives, our approach to Quality Assurance defines the principles, standards, and procedures that provide the foundation for our project activities. For Covansys, Quality Assurance is an integral component of our Quality Management methodology.

Quality Management

The QA Team places a high emphasis on quality management, incorporating it up front into the planning process and ensuring that it is an integral part of the deliverable production process, with multiple measurement points. Quality management is a continuous process that incorporates the three processes of quality planning, quality assurance, and quality control.

For the NJ SVRS project, quality management involves the processes required to ensure that the project meets or exceeds requirements and expectations of the stakeholders and the end-user community. During the planning phase, the project team agrees on quality and performance metrics used as a measure for continuous project success. The major processes by which we achieve quality on a project are quality planning, quality assurance and quality control.

- **Quality Planning** – The initial step is the development of a Quality Assurance Plan. This document represents this plan. By developing this plan, project quality standards and project guidelines for each phase and deliverable of the project are identified up front, before work begins.

- **Quality Assurance** – The QA Team conducts quality assessments, using a dedicated QA staff, on a regular basis to evaluate the overall performance of the project and to ensure that the project is satisfying pre-defined quality standards.

- **Quality Control** – For the NJ SVRS project, the QA Team continuously monitors specific project results and identifies ways to reduce and/or eliminate unsatisfactory results. Utilizing quality checkpoints, incorporated into the Quality Assurance Plan, the QA Team facilitates product walkthrough, inspections, and reviews against the standards documented for each deliverable. It is the responsibility of the Covansys Project Manager to educate the project team on the Quality Assurance Plan and other relevant project guidelines and to hold them accountable for conformance to the quality standards.

Peer review definition

Covansys defines Peer Review Management as the act of planning, implementing, monitoring, and tracking the collective set of activities, tasks, and processes necessary to identify and remove defects from work products as early and efficiently as possible. A Peer Review is a thorough review of work products to identify defects and areas for improvement. The reviews are conducted by the peers of those who produced the work products.

Peer Reviews activities consist of the following tasks:

Conduct Peer Review

Document and Communicate Findings

Track Action Items to Completion

Covansys applies quality processes and tools (such as the Capability Maturity Model) for conducting successful peer reviews. The deliverables and work products that may be reviewed include, but are not limited to, software requirements, software design, code, and test procedures. The Peer Review Procedure will specify which of the project's work products and deliverables should undergo peer review, the schedule for the reviews and which Project Team Members are to be involved.

Peer review team

Most peer reviews consist of a minimum of 3 project team members: the session facilitator, the project team member who created/developed the materials under review, and the key reviewer.

For more complex deliverables or work products the team would consist of 4 members, the fourth being an additional key reviewer. All participants are peers. Covansys discourages project managers or supervisors, of the team member whose material is under review, from participating in order to prevent the session from becoming a ‘performance review’ rather than a ‘peer review’. However, there are exceptions based upon the critical nature of the deliverable or work product.

Prior to the review, one team member will be identified as the facilitator or person responsible for conducting the peer review. The facilitator will distribute materials to the reviewers so they can prepare for the peer review. Materials that will be distributed include those that describe the objective of the work product, applicable standards, and relevant supporting documentation (e.g., the design for code that is to be reviewed.).

Procedures and checklists

At the beginning of the session the facilitator will review the ‘ground rules’ of the session with the participants and distribute the checklist of items to be covered. The checklist will include the objective of the review, a list of programs/modules/work products to be reviewed, the current state of materials (i.e., draft, 1st review, 2nd review, final) and the questions to be answered during the course of the review. In addition, free text space is provided in which to insert comments or concerns encountered during the review. At the end of the session the peer review facilitator will record the results using the Peer Review Summary sheet.

This sheet includes documenting:

- Date of review
- Participants
- Products reviewed
- Number and types of defects identified
- Suggestions for correcting defects
- Other action items that must be accomplished
- Re-review date or target date for correction of defects
- Actual completion date
- Participant notes (attached to summary)

The facilitator, along with the other reviewer(s), will determine if another review is necessary. Documented reviews will be placed in the project notebook maintained by the Covansys Project Manager.

Statistics compiled

Each potential or real issue discovered during the course of the peer review walkthrough will be documented based on its criticality to application functions. The issues will be assigned a size and complexity rating as well as a priority ranking as a way to understand the effort that will be required to correct the deficiencies and the necessary schedule that must be in place in order to meet project deliverable timeframes. The statistics captured include:

CONFIGURATION MANAGEMENT

Control of changes to requirements, design, and code

Covansys defines Configuration Management to be the discipline of identifying, documenting, reporting, and tracking changes that may occur to any aspect of a project (i.e., scope, time, or cost) throughout the life cycle of a project. This includes Software Configuration Management ("SCM"), which deals specifically with changes to the software product itself whether from changed requirements, or to support version control or to support changes to environmental factors affecting operational software or networks.

Covansys and the State understand that the spirit of change management is to communicate, document and control the impact of changes to project scope on level of effort, cost, and deadlines. Although some changes may be informal in nature, they still must follow the change control process. Changes to project scope (i.e., requirements, design, code, interfaces, etc.) will be managed through a change control process that will involve investigation, documentation, review and approval, deferral, or rejection of a change request. Please refer to the Consolidated Management Plan under Change Management for a description of the change control procedures.

Any changes to election laws or statutes that are specific to an individual State are handled through the same change control request process. A State, at any time, can request the product to be upgraded due to a change in the State's election law. Based on the State's request, a thorough gap analysis will be conducted and based on the findings a formal Change Control Request proposal will be submitted. Based on the State's approval of the proposal the change to the product will be programmed and delivered according to a mutually agreeable schedule.

Control of interface changes

Controlling and managing change to requirements, design, and code to the application proper equally applies to the necessity of similar changes for interfaces. Therefore, Covansys will apply its change management process for interfaces as outlined in the Consolidated Management Plan.

Tools to control versions and builds

Covansys recognizes that the ability to control software versions, their builds, migrations, and eventual deployment is a critical success factor to the project. Our approach to SCM assists in the establishment and maintenance of the integrity, pertaining to software products. To achieve that goal Covansys and its software partner, PCC Technology, employ the use of Microsoft's Visual SourceSafe for all version and migration control. By providing project-oriented software management, Visual SourceSafe enables the Covansys team to develop applications knowing that their projects and files will be protected. Several features of Visual SourceSafe include:

- Automatically protects and tracks source code, documentation, binaries, and all other file types as they change throughout the software life cycle.
- Full check in and check out features that securely protects files from accidental overwrite by preventing more than one user from modifying the same file at once.
- Versioning features provide snapshots of a project for the quick retrieval of any previous version in the software life cycle.
- Difference reporting provides quick access to changes across separate versions of the same file, enabling developers to know immediately what lines of code have changed.
- Allows team members to reconcile conflicts between different versions of the same file by using a visual merge capability, which provides a point-and-click interface for uniting files and avoids potential loss of valuable changes.

Parameters for regression testing

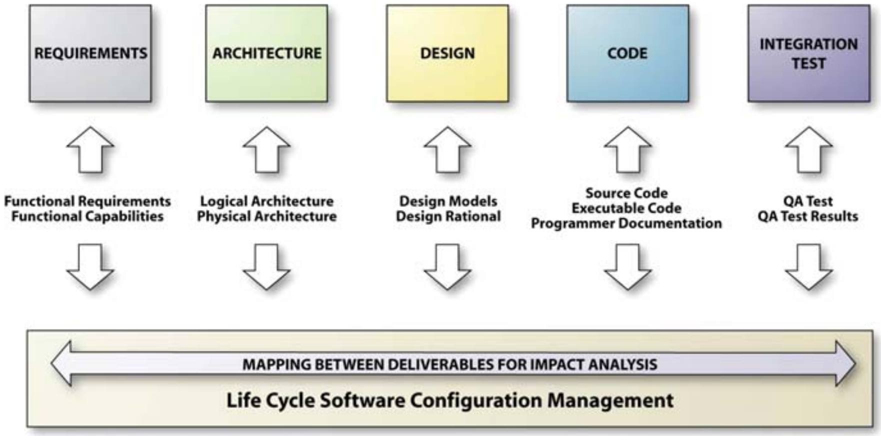
Integration test is a key component of the Project Life Cycle Configuration Management Model. Included in this component are elements necessary to appropriately plan, monitor, and control regression testing. Nearly every 'fix or change' requires a certain degree of re-testing. The degree and nature of re-testing is based upon parameters that are a function of the relationship that exists between the components and modules that comprise the software product. The Covansys team has established the following guidelines (i.e., identifying characteristics of critical relationships) for determining the parameters for regression testing:

- Disjoint – Do modules/components exist in a mutually exclusive relationship?
- Hierarchical – Is there a hierarchical structure that demands a top-down or bottom-up cycle of testing?
- Intersection – Does module/component functionality overlap to the degree that common touch points must generate identical results?
- Union – Do individual modules/components function as a single unit in order to complete a single task?
- Independence – Does the occurrence of a particular action in one module/component affect the result (outcome) of actions taken in another module?

Once these questions are answered, parameters are then set regarding testing. Briefly, the answers will dictate the nature and extent of screen navigation paths, action buttons, process flows, business rules and edits (both field-level and application level) to re-test in order to ensure system completeness and integrity.

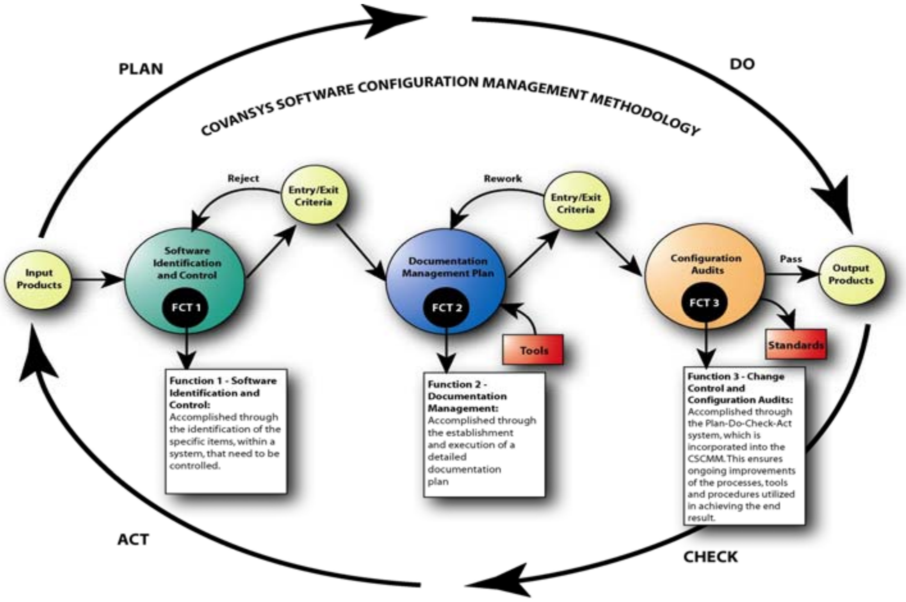
Baselines established for tools, change logs, and modules

At Covansys, we understand Software Configuration Management (SCM) to be the discipline of identifying the configuration of a system at discrete points in time, for systematically controlling changes to this configuration and maintaining the integrity and traceability of this configuration throughout the life cycle of the software system.



Our approach, achieves a one-to-one mapping with the three functions of the Configuration Management Model, in relation to information systems.

Covansys Software Configuration Management Methodology



Function 1 - Software Identification and Control: Accomplished through the identification of the specific items, within a system, that need to be controlled.

Function 2 - Documentation Management: Accomplished through the establishment and execution of a detailed documentation plan

Function 3 - Change Control and Configuration Audits: Accomplished through the Plan-Do-Check-Act system, which is incorporated into the CSCMM. This ensures ongoing improvements of the processes, tools and procedures utilized in achieving the end result.

The approach, based on the Software Engineering Institute (SEI) model, focuses on three areas:

- **Software Identification and Control:** Through this component, an identification scheme is developed by which the structure of the product is reflected. The identification function is addressed through identifying the different and unique application components, giving each a name, a version identifier and a configuration identifier. The control function is achieved through controlling releases of the product throughout the life cycle, by having controls in place that ensure consistent software via the creation of a baseline product.
- **Documentation Management Plan:** Through this component, documents are identified, tracked, traced, versioned, and updated throughout the product lifecycle. This component in particular lends itself to accountability as it deals with the recording and reporting of the status of components, including documents, and change requests. It is also instrumental in gathering vital statistics about the individual modules that comprise the end product. As noted above, Visual SourceSafe includes full documentation features. Visual SourceSafe securely provides an audit trail for every file and every project for all changes. This includes the check-in/check-out process.
- **Configuration Audits:** This aspect validates the completeness of the product and maintains consistency among the various components by ensuring that they exist in

a readily accessible and acceptable state throughout the entire project lifecycle. Ultimately, the product is a well-defined collection of these individual components.

Change Request Process

Change control is a very methodical process and, as such, necessitates the need for thorough and accurate documentation and record keeping. Covansys' methodology, as described in the Consolidated Management Plan, provide for the required accountability sought by the State of New Jersey.

Change Control Board and Change Control Process

Please refer to the Consolidated Management Plan for the functions of the Change Control Board and a description of the change control process.

Error tracking

The peer review facilitator is responsible for tacking the action items identified in a peer review until they are accomplished. The results of the peer review session will serve as a type of issue log from which the tracking will occur. Based on the target date for completion of corrections, the facilitator will meet with the individual team member responsible for correcting the defects. At that time a one-on-one walkthrough will be conducted of each item identified in the Peer Review Summary sheet as requiring correction. The team member must provide evidence of successful completion of corrections. This evidence may include but is not limited to:

- Demonstrated functionality
- Before/after images of code
- Diagrams, models, charts
- Successful test results

The Peer Review Summary sheet will be updated based on this evidence. Any items which have not been completed by the agreed upon date will be escalated to the Covansys Project Manager.

DEVELOPMENT PLAN MILESTONES

The project plan for the New Jersey SVRS provides for the following timetable of the major steps in the Design and Configuration processes for the SVRS:

Milestone	Estimated Completion	Actual Completion
Completion of JAD sessions		April 15, 2005
FRSD Approval	May 2, 2005	June 17, 2005
RTM Approval	May 11, 2005	June 13, 2005
Delivery of Preliminary Test Scripts	July 7, 2005	
Deliver UAT Schedule	June 24, 2005	June 24, 2005
Complete Development Phase - 1 Modules and Internal Testing Voter Registration Maintain County Data Elections System Inquiries Maintain Voter History Duplicate Voter Polling Place Absentee Ballots Reports 1 Letters 1	August 17, 2005	
Start UAT Phase 1	August 22, 2005	
Complete Development Phase - 2 Modules and Internal Testing Letters 2 Reports 2 Petitions Candidate Management Poll Worker External Interfaces Redistricting County Backup Tasks	September 12, 2005	
Start UAT Phase 2	September 12, 2005	
Completed UAT Phases 1 and 2	September 26, 2005	
Complete System Performance Testing	September 26, 2005	
Receive State Sign-off on Test Results	September 30, 2005	

COLLECTION OF REQUIREMENTS

During the Business Needs Assessment/Gap Analysis phase, the project team gathers business, functional, and technical requirements through a combination of surveys and requirements sessions (referred to as Joint Application Development (JAD) sessions) with the State's Subject Matter Experts (SMEs), eventual users of the system, and the key stakeholders. The Functional Lead Analyst guides and facilitates these sessions. Participants review components and modules, walking through each **ElectioNet™** screen and function, which comprises the core product, with the sole purpose of gathering and validating State specific business requirements. In addition, the team will solicit information from the participants as to the business rules supporting how the requirements apply to New Jersey.

After collecting these requirements, an analysis determines the "gaps" that may exist between the **Electio*Net*** application functionality and New Jersey's requirements. The Gap Analysis document captures the gaps in the application serve as the basis for the application configuration as well as a clearinghouse for gap-related issues. The team maintains a gap status including a measure of priority, impact, relevant details about the gap, and resolutions pertaining to how best to 'close' the gap. This information along with the solicited feedback will be critical in completing the analysis and determining the best application design appropriate for New Jersey.

ANALYSIS AND DESIGN

Once the Gap Analysis document, representing the require changes, are reviewed and accepted by the State, the team develops a conceptual model encompassing detailed designs for the proposed operational solution. Detailed descriptions and modeled representations, where applicable, will be developed and documented to describe how the system will support each identified NJ SVRS functional requirement.

The Functional Requirement Specification Document (FRSD) defines the functional requirements. The team, lead by the business analysis in concert with the application developers, prepares the FRSD. The FRSD details the features and behavior of the target custom application, based on State-specific requirements together with the already resident HAVA requirements. The FRSD includes processing logic/flow, business rules, preliminary screen designs, action buttons, and navigation for those screens requiring changes, as well as including a more complete level of functional detail for areas pertaining to identified gaps.

To serve its purpose of driving application changes as well as determining test cases and scenarios that will test every aspect of the application, the FRSD will clearly cover the functionality of the entire application. The team documents key design issues identified while conducting these activities as part of the effort in addition to having been already communicated to the State via the regularly scheduled project status meetings. Working with the State, the team will seek reasonable solutions to any design issues to ensure system acceptance and success.

By documenting these issues as a function of the FRSD, the State will have thorough documentation of what was discussed, what was approved and what was deferred along with reasons should the need arise to revisit any or all decisions. Ultimately, this document will be essential in maximizing the commonality between cities and towns with respect to use of the software, which will be critical to the success of the project.

At the same time as the FRSD preparation, database designers are reviewing the Gap Analysis document and gathered business requirements in order to design the necessary changes to the **Electio*Net*** physical database schema necessary to accommodate New Jersey. Entity-Relationship diagrams will be prepared depicting the required changes and containing descriptions of each change.

The required database design maps the business rules and logic included and supported in the FRSD. This will address required features currently supported by the **Electio*Net*** solution as well as new feature support, with direct traceability back to the requirement documents.

APPLICATION BUILD / CONSTRUCTION

Once all changes to requirements and/or designs have been approved by the State, the Covansys application development team will commence with construction/modifications. The developers rely on FRSD documents to ensure all state specific requirements are coded and integrated into the base code for each ElectioNet module. All development must adhere to strict version control and all code is quality checked via peer and code reviews, documenting recommended changes and providing an audit trail for ensuring quality standards are maintained

TESTING

Internal Testing

The Covansys Team's method for testing involves multiple tasks.
Although these tasks vary in importance as the software development lifecycle unfolds, it is important to ensure a balance of focus is retained across these different work efforts. During the software modification phase, the Covansys team will conduct a variety of tests including:

- Developer Test
- Unit Test
- System Test
- Integration Test

Developer Test – The phrase "Developer Testing" is used to categorize the testing activities most appropriately performed by software developers. It also includes the artifacts created by those activities. Developer testing encompasses the work traditionally thought of under the following categories: Unit Testing, much of Integration Testing, and some aspects of what is most often referred to as System Testing. Even within the scope of a single iteration, it is usually more efficient for the developer to find and correct as many of the defects in their own code as possible, without the additional overhead of handing it off to a separate test group. The desired result is the early discovery of the most significant software errors – regardless of whether those errors are in the independent unit, the integration of the units, or the working of the integrated units within a meaningful end-user scenario.

Unit Test – Unit Test is a more traditional distinction that focuses on verifying the smallest testable elements of the software. Unit testing is typically applied to components in the implementation model to verify that control flows and data flows are covered and function as expected. These expectations are based on how the component participates in executing a use case, which you find from sequence diagrams for that use case. The Implementer performs unit test as the unit is developed.

System Test – System testing denotes the aspects of the test effort that are most appropriate for a group independent of the software developers to undertake. The target is the typically the end-to-end functioning of the system. System testing typically has several phases that include:

- Business Function Testing – to ensure that the business needs have been met
- Integration Testing – to ensure that the various modules of the system work
- seamlessly
- Load/Stress Testing – to ensure that the system can meet the performance
- requirements
- System Security Testing – to ensure that the software meets the security requirements

Integration Test – Integration testing will be part of system testing and is performed by the Quality Assurance (QA) test team. The integration testing will validate the overall design and architecture of the software solution within the target environment. This will include testing the movement of data through all interfaces, ensuring the platform is stable and ensuring the code is stable. The team will work with the DoE’s Office to appropriately define the test plans and scripts based on State, HAVA, and County/municipalities requirements. Covansys will expressly designate a person from this team as a Liaison to the DoE’s Office during the acceptance testing period.

External Testing – User Acceptance Testing

User Acceptance Testing (UAT) begins after all defects identified during the various phases of internal testing have been repaired. UAT test scripts will be authored by State project personnel with assistance from Covansys. To facilitate this process, Covansys will provide the State will all test scripts used during internal testing. The State may decide to reuse, modify, or not use any or all of the provided scripts. The focus of UAT is for the State personnel to execute test scripts that accurately simulate the way ElectioNet will be used in production. It will be critical for New Jersey's UAT to include scenario's reflecting real world system use at both the state and county levels. In addition, Covansys will work with the New Jersey team to ensure that every type of system role for both the state and county election officials are represented by at least one member of the UAT team. To expedite the UAT process, Covansys will be delivering the various ElectioNet modules in two groups referred to as Phase 1 and Phase 2. This will allow an extended internal test of the modules prior to UAT and will stagger the delivery of modules to avoid overwhelming the UAT testers. The end result will be less defects going into UAT and a more manageable amount of code being introduced to the testers at one time.

Covansys will be providing the UAT testers with access to an automated, web based tool to track test script execution and results. In addition, this tool will provide the New Jersey team with reports relative to progress on test script execution, number of defects reports, defects resolved, Etc. By the conclusion of the UAT period, Covansys will have repaired all identified defects and New Jersey's ElectioNet system will be ready for rollout to the pilot counties.

ATTACHMENT 1

Development Methodology

Focus		Project Delivery				
Phase		Requirements	Analysis & Design	Build	Test	Implement
Activities		Begin identifying New Jersey specific requirements via JAD and GAP sessions. Obtain state approval for JAD and GAP documents. Translate JAD and GAP documents into draft FRSDs.	Develop detailed requirements for the solution. Create detailed plans for testing and end user training. Finalized estimates into a committed delivery date and cost.	Build ElectioNet modules to New Jersey specs and conduct Unit Testing and Pre-Integration Testing. Develop materials that will be used in training of end users. Develop test cases for all subsequent testing stages.	Conduct all subsequent testing stages as defined in the Testing Strategy in preparation for final implementation. Provide end users with training. Develop detailed plan for transition of solution from project team to support team.	Conduct Pilot of solution is applicable. Fully deploy solution to end users. Transition solution from project team to support team. Close out project.
DELIVERABLES	Requirements Related	JAD & GAP Documents Produced and Reviewed	Functional and Technical requirements finalized and documented in: FRSD, RTM, TADD, DBDD	New requirements conveyed via CCR process.	New requirements conveyed via CCR process.	New requirements conveyed via CCR process.
	Development Related		Technical and Functional Designs are locked.	Pre-Integration tested solution, most errors detected and fixed	Integrated Module Internal and UAT Testing. Progressively fewer errors detected and fixed	Minor, final defects identified and fixed.
	Time / Cost Related		Development plan & timetable, CCR's identified and approved.	Data Conversion completed and loaded to ElectioNet	Data replication & backup from County's tested	Local Servers live in every county
	Quality Related	Internal Document Review, Obtain State Signoff	Internal Document Review, Obtain State Signoff	Test Cases, Scripts, and Data	Internal & User Acceptance Test Results	Pilot Results
	Training Related	Training Strategy	Training Plans	Training Materials	User Training for UAT Testers	Just in Time Training prior to rollouts to each county
	Implementation Related		Full Implementation & Technical Implementation plan completed	Pilot & Production Transition Plan Completed	Technical Implementation - Plan	Project moves to Warranty Phase
	Project Management Related	Status Reports, Project Plan, Issue and Decision Log, Risk Management Plan, Communication Plan, Resource Plan	Status Reports, Project Plan, Issue and Decision Log, Risk Management Plan, Communication Plan, Resource Plan	Status Reports, Project Plan, Issue and Decision Log, Risk Management Plan, Communication Plan, Resource Plan	Status Reports, Project Plan, Issue and Decision Log, Risk Management Plan, Communication Plan, Resource Plan	Status Reports, Project Plan, Issue and Decision Log, Risk Management Plan, Communication Plan, Resource Plan
Reporting Milestones		JAD and GAP sessions & documents -Complete	Analysis & Design Phase - Complete, Tech & Functional Design Complete, Development & Implementation Plan Complete. Signoff on FRSD & RTM	Build Phase Complete, Initial Internal Testing Complete, UAT Test Scripts Complete	User Acceptance Testing Completed Test Phase Complete	Final Delivery Signoff Implementation Phase after Full Rollout,
Major Check Points		Agree to Proceed	Agree to Proceed	Agree to Proceed	Agree to Proceed	